sensors

MDPI

Article

Scale-Aware Tracking Method with Appearance Feature Filtering and Inter-Frame Continuity

Haiyu He [1], Zhen Chen [1], Zhen Li [1], Xiangdong Liu [1] and Haikuo Liu [2,*]

[1] School of Automation, Beijing Institute of Technology, Beijing 100010, China; 3220195103@bit.edu.cn (H.H.); chenzhen76@bit.edu.cn (Z.C.); zhenli@bit.edu.cn (Z.L.); xdliu@bit.edu.cn (X.L.)
[2] School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100010, China
* Correspondence: foreverlhk@bit.edu.cn

Abstract: Visual object tracking is a fundamental task in computer vision that requires estimating the position and scale of a target object in a video sequence. However, scale variation is a difficult challenge that affects the performance and robustness of many trackers, especially those based on the discriminative correlation filter (DCF). Existing scale estimation methods based on multi-scale features are computationally expensive and degrade the real-time performance of the DCF-based tracker, especially in scenarios with restricted computing power. In this paper, we propose a practical and efficient solution that can handle scale changes without using multi-scale features and can be combined with any DCF-based tracker as a plug-in module. We use color name (CN) features and a salient feature to reduce the target appearance model's dimensionality. We then estimate the target scale based on a Gaussian distribution model and introduce global and local scale consistency assumptions to restore the target's scale. We fuse the tracking results with the DCF-based tracker to obtain the new position and scale of the target. We evaluate our method on the benchmark dataset Temple Color 128 and compare it with some popular trackers. Our method achieves competitive accuracy and robustness while significantly reducing the computational cost.

Keywords: discriminative correlation filter; scale estimation; color name; salient feature; visual tracking

Citation: He, H.; Chen, Z.; Li, Z.; Liu, X.; Liu, H. Scale-Aware Tracking Method with Appearance Feature Filtering and Inter-Frame Continuity. *Sensors* **2023**, *23*, 7516. https://doi.org/10.3390/s23177516

Academic Editor: Ikhlas Abdel-Qader

Received: 7 July 2023
Revised: 11 August 2023
Accepted: 25 August 2023
Published: 30 August 2023

1. Introduction

Visual object tracking is a fundamental research topic in computer vision and pattern recognition that has many applications, such as robot vision [1,2], video surveillance [3–5], medical–industrial integration [6–8], etc. The main task of visual tracking is to estimate the target's position and scale according to the given target specified by a bounding box in the first frame. Trackers need to achieve both high accuracy and efficiency to be feasible for many applications [9]. However, visual object tracking faces many challenges that affect its performance and robustness. Scale variation, which occurs when the target changes its size or distance from the camera, is one of the most difficult challenges that affects the performance of trackers [10,11].

DCFs are effective trackers that achieve high accuracy and efficiency by transforming the matching process from the spatial domain to the frequency domain [12]. However, DCF cannot estimate the scale change [13] by itself. Since DCF-based methods are based on templates used to locate objects, long image processing time will lead to tracking failure due to boundary effects [14]. Existing scale estimation methods based on multi-scale features are computationally intensive and degrade the real-time performance of DCF-based trackers, especially in scenarios with limited computational power [15].

In this paper, to address the above issues, we propose a computationally efficient and scale-adaptive tracking method based on DCF and probability estimation. The main contributions of this paper can be summarized as follows:

(1) We propose a practical scale-adaptive tracking method based on probability estimation (PSACF), which can handle scale changes without using multi-scale features and

can be combined with other tracking methods as a plug-in. We also adopt a fusion tracking strategy to fuse the results with the position localization results.

(2) We propose a local adaptive salient feature and scale restoration method, which selects the local salient features of the target in the current image and reduces the interference caused by background clutter. Based on global and local scale consistency assumptions, we perform global scale recovery based on the Gaussian model and inter-frame continuity of salient features, which improves the accuracy and robustness of scale estimation.

The rest of this paper is organized as follows. Section 2 introduces related works, Section 3 revisits the related formulation, and Section 4 presents the detailed design. Experiments are conducted to verify the proposed algorithm in Section 4. Finally, Section 6 gives the main conclusions.

2. Related Works

Visual tracking has been studied extensively over the past decades. In this section, we discuss the methods closely related to this work in terms of DCF-based trackers and scale estimation methods.

2.1. DCF-Based Trackers

DCFs are effective trackers that achieve high accuracy and efficiency by transforming the matching process from the spatial domain to the frequency domain. During the past decades, a variety of tracking algorithms based on DCF have been proposed [10,13,16–19]. The minimum output sum square error (MOSSE) tracker proposed by Bolme et al. [20] showed the strong potential of DCFs, whose processing speed reached 669 FPS (frames per second) . However, MOSSE only utilized gray features, which limited its performance in complex scenarios. To address this limitation, Henriques et al. [13] exploited the cyclic structure with a kernel (CSK) based on MOSSE. By doing so, they extended the gray features to multichannel HOG features and proposed the kernel correlation filter (KCF), which improved the tracking performance while maintaining a high processing speed of 172 FPS. However, both MOSSE and KCF suffered from boundary effects, which caused unwanted background information to be included in the target model. To address this problem, Danelljan et al. [16] proposed spatially regularized DCFs (SRDCF), which introduced a spatial regularization term to penalize filter coefficients corresponding to background regions. SRDCF achieved superior performance over KCF, but at the cost of reduced speed.

One of the problems of DCF-based trackers is that they cannot handle scale variation by themselves. In order to estimate the target scale, they need to use additional methods or components, which we review in the next subsection.

2.2. Scale Estimation Methods

Scale variation is one of the most challenging problems in visual tracking. A tracker that cannot handle scale variation may lose the target or drift to the background when the target changes its size or distance from the camera.

One common way to estimate the scale is to exploit multi-scale features. For example, discriminative scale space tracker (DSST) [10] and scale adaptive with multiple features tracker (SAMF) [11] trained a separate scale classifier by constructing a multi-scale image pyramid for the detection of target scale variation. They first detected the target position by DCF, and then extracted multi-scale features in the predicted position regions to estimate the scale using another DCF. However, these methods were computationally expensive and slowed down the tracker, especially in scenarios with restricted computing power.

Another way to estimate the scale is to traverse the scale space to match the target or train a scale classifier. Ma et al. [21] proposed a scale searching scheme that focused on the response map and incorporated the average peak-to-correlation energy criterion into a multi-resolution translation filter framework to handle scale variation. Lu et al. [22] applied an additional correlation filter over scale-aware region proposals for scale estimation. They built a region proposal algorithm based on EdgeBox by adopting a support vector machine

(SVM) classifier to obtain a set of proposals for scale and position detection. Zhang et al. [18] proposed a multitask correlation particle filter for robust visual tracking. They adopted a particle sampling strategy to address the scale variation issue. Vojir et al. [23] proposed scale-adaptive mean-shift (ASMS), in which they adopted a scale estimation method and achieved a relative balance between the tracking accuracy and computational efficiency. There are also some methods that use deep neural networks to tackle the scale variation problem [24–27], but deep learning-based methods heavily rely on GPUs and are difficult to apply in scenarios with limited computing resources.

However, these methods still relied on some form of multi-scale feature or multi-scale traversing strategy, which increased the computational cost and complexity of the tracker. Moreover, they did not consider the inter-frame continuity or consistency of appearance between frames, which could significantly reduce the computation cost. We exploit the distribution of the salient feature of the appearance and employ the probability estimation method in this paper to tackle the scale variation problem. Our method achieves competitive accuracy and robustness while significantly improving the processing speed over existing methods.

3. Discriminative Correlation Filter

In this paper, the KCF [13] is adopted for its efficiency. In this section, we briefly review the KCF tracking method with the fundamental formulas. The proposed scale estimation method is detailed in the next section.

The core idea of the DCF algorithm is to train a classifier via the ridge regression method by minimizing the squared error over the samples x_i and their regression targets y_i,

$$\min_{w} \sum_{i} (f(x_i) - y_i)^2 + \lambda \|w\|^2. \tag{1}$$

The minimizer has a closed form, which is given by

$$\mathrm{w} = \left(X^H X + \lambda I\right)^{-1} X^T y, \tag{2}$$

where X is the data matrix, and X^H is the Hermitian transpose. For nonlinear regression, the kernel trick is applied to the model

$$\mathrm{w} = \sum_{i} \alpha_i \varphi(x_i), \tag{3}$$

where α is the vector of coefficients α_i, and $\varphi(x_i)$ is the arbitrary nonlinear mapping of x_i. The filter can be accelerated via discrete Fourier transform (DFT) as

$$\widehat{\boldsymbol{\alpha}} = \frac{\widehat{\boldsymbol{y}}}{\widehat{\boldsymbol{k}}^{xx} + \lambda}, \tag{4}$$

where $\widehat{\boldsymbol{k}}^{xx}$ is the cross-correlation in the high-dimensional space $\varphi(x)$. In this paper, the Gaussian kernel is adopted as

$$\boldsymbol{k}^{\mathrm{xx}'} = \exp\left(-\frac{1}{\delta^2}\left(\|x\|^2 + \|x'\|^2 - 2\mathcal{F}^{-1}\left(\widehat{\boldsymbol{x}} \odot \widehat{\boldsymbol{x}}'^{*}\right)\right)\right), \tag{5}$$

where $\mathcal{F}^{-1}$ denotes the inverse FFT transform. The response in the Fourier domain can be calculated as

$$\hat{f}(\boldsymbol{z}) = \left(\widehat{\boldsymbol{k}}^{\boldsymbol{xz}}\right)^{*} \odot \widehat{\boldsymbol{\alpha}}. \tag{6}$$

When transformed back into the spatial domain, the response map denotes the location of the target being tracked.

4. Proposed Algorithm

In this section, an introduction to our tracker is briefly given. Additionally, we present the overall process of our salient feature filtering and scale estimation scheme in detail. Finally, the mechanism of result fusion and the strategy of template updating are discussed. An overview of the proposed method is given in Figure 1, and the pseudo-algorithm is shown in Algorithm 1.

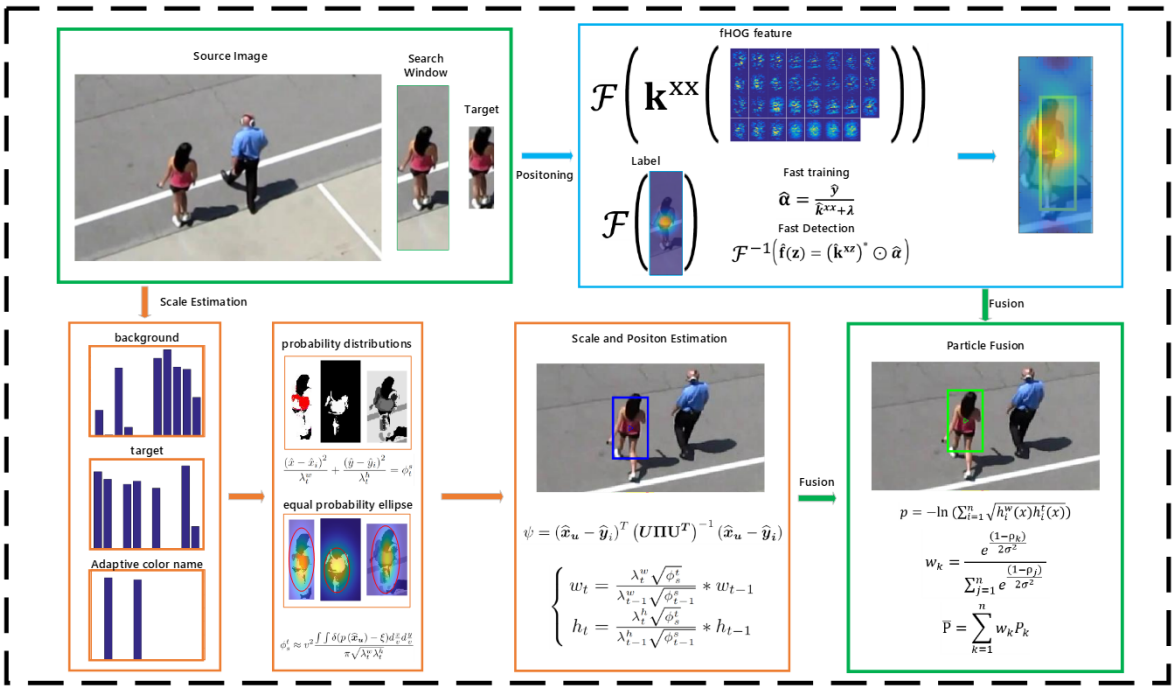


Figure 1. Overview of the proposed method.

Algorithm 1 PSACF: Probability Estimation and Scale-Adaptive Correlation Filter Tracking

Input: The first frame I_1 and the target region R_1
Output: The target region R_t in each frame I_t

1: Initialize the target appearance model H^t and the background appearance model H^b based on the CN feature
2: Initialize the target position P_1 and scale S_1 based on the target region R_1
3: **for** t = 2,3,... **do**
4: Extract the search window W_t around the target position P_{t-1} in the frame I_t
5: Calculate the salient feature H^t of the target based on Equation (10)
6: Estimate the probability $p(\hat{x}, \hat{y})$ of each candidate pixel in the search window based on Equation (15)
7: Weight the probability $\hat{p}(\hat{x}, \hat{y})$ of each candidate pixel based on Equation (17)
8: Estimate the scale S_t of every salient feature based on Equations (24) and (26)
9: Update the target appearance model H^t based on Equation (32)
10: Obtain two position candidates P_{tc} and P_{tk} from the proposed algorithm and the KCF algorithm, respectively
11: Calculate the weights w_{tc} and w_{tk} of each position candidate based on Equation (31)
12: Fuse the position candidates to obtain the final position estimate $P_t = w_{tc}P_{tc} + w_{tk}P_{tk}$
13: Output the target region $R_t = (P_t, S_t)$ in the frame I_t
14: **end for**

4.1. Fast Adaptive Salient Feature Filtering Algorithm

In this paper, the CN feature is adopted as the appearance feature. The CN features are based on a set of 11 basic color names: black, blue, brown, gray, green, orange, pink, purple,

red, white, and yellow. Each pixel in the image is assigned a probability distribution over these 11 color names based on a probabilistic model trained on a large dataset of natural images. The model used is provided by [28].

The CN histogram includes 11-dimensional features, most of which are invalid or are interference items from the background. During tracking, these invalid features in the background will interfere with the target. To reduce the influence of these features, the weight of these features in the target histogram needs to be suppressed or deleted. In the process, three histograms are involved, i.e., the feature histogram of the target, the feature histogram of the background, and the salient feature histogram of the target.

Firstly, the histogram of the target H^t and the histogram of the whole search window H^a are modeled. Additionally, the background histogram H^b is defined. It is obvious that the histogram of the whole search window is divided into the arithmetic sum of the target histogram and the background histogram:

$$H^t = \left[\begin{array}{cccc} h_1^t(x), & h_2^t(x), & \cdots, & h_1^t(x) \end{array} \right], \tag{7}$$

$$H^a = \left[\begin{array}{cccc} h_1^a(x), & h_2^a(x), & \cdots, & h_n^a(x) \end{array} \right], \tag{8}$$

$$H^b = H^a - H^t. \tag{9}$$

The back-projection is then applied to the search window in order to obtain the target candidate probability distribution. Accordingly, the target histogram is rebuilt by

$$\hat{h}_i^t(x) = C \sum_{i=1}^{n} \frac{\ln\left(\frac{h_i^a - h_i^t + 1}{h_i^t + 1}\right)}{(x_i - x_c)^2} h_i^t(x_i) \mathcal{M}(x_i), \tag{10}$$

where $h_i^t(x)$ denotes the i-th bin of the salient feature histogram of the target, x_i denotes the pixel position, and x_c is the center of the search window. n is the number of pixels in the search window, $h_i^a(x)$ is the i-th bin of the search window, C is a normalization constant, and $\mathcal{M}(x)$ maps the value of the pixel at location x to the corresponding bin in the CN space. The above process can be regarded as a weighting process. $(x_i - x_c)^2$ denotes the distance between the pixel location and the center of the search window, and we utilize this distance to lower the importance of edge pixels. $h_i^t(x_i)$ represents the probability that the pixel at x_i belongs to the target.

Therefore, the salient feature histogram of the target can be further described as:

$$\widehat{H}^t = \left[\begin{array}{cccc} \hat{h}_1^t(x), & \hat{h}_2^t(x), & \cdots, & \hat{h}_n^t(x) \end{array} \right]. \tag{11}$$

In some application scenarios, there is no solution to (10). As a result, we choose the last two bins as the salient features.

4.2. Scale Estimation

The candidate pixel at $\widehat{\boldsymbol{x}}_{\boldsymbol{u}} = (\hat{x}, \hat{y})^T$ is defined as $I(\hat{x}, \hat{y})$. It is worth mentioning that there still exists a partial background in the target bounding box, which causes interference in scale estimation. Therefore, the probability $p(\widehat{\boldsymbol{x}}_{\boldsymbol{u}})$ of whether $I(\hat{x}, \hat{y})$ belongs to the target needs to be estimated:

$$\hat{q}_u = C \sum_{i=1}^{n} K\left(\|x_i\|^2\right) \delta(h(x_i) - u), u = 1, \cdots, m, \tag{12}$$

where δ is the Kronecker delta, and i denotes the pixel location. $h(x_i)$ is the corresponding bin in the feature space of the i-th pixel, $K\left(\|x_i\|^2\right)$ is the RBF kernel function [23], and C is a normalization constant.

According to Formula (12), the probability can be estimated as follows based on the Bayesian formula

$$p(\hat{x},\hat{y}) = p\left(I \in H^t \mid I \notin H^b\right) = \frac{p\left(I \in H^t\right) * p\left(I \notin H^b \mid I \in H^t\right)}{p\left(I \notin H^b\right)}, \tag{13}$$

where $I \in H^t$ denotes the event that pixel I belongs to the target, and $I \notin H^b$ denotes the event that I does not belong to the background in the search window.

When screening for salient features, the background features in the search window have been suppressed; therefore,

$$p\left(I \notin H^b \mid I \in H^t\right) = \frac{p\left(I \notin H^b, I \in H^t\right)}{p(I \in H^t)} = \frac{p\left(I \in \widehat{H}^t\right)}{p(I \in H^t)}, \tag{14}$$

and Formula (13) can be described as

$$p(\hat{x},\hat{y}) = \frac{p\left(I \in H^t\right) * p\left(I \notin H^b \mid I \in H^t\right)}{p\left(I \notin H^b\right)} = \frac{p\left(I \in \widehat{H}^t\right)}{1 - p\left(I \in H^b\right)}. \tag{15}$$

Meanwhile, the farther the pixel is from the center, the lower the probability that it belongs to the target; thus, we define the distribution of candidate pixels in the search window as a Gaussian model:

$$N(\widehat{\boldsymbol{x}}_{\boldsymbol{u}}, \widehat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{\Psi}) = \frac{1}{\boldsymbol{\Phi}} \exp\left(-\frac{1}{2}(\widehat{\boldsymbol{x}}_{\boldsymbol{u}} - \widehat{\boldsymbol{y}}_{\boldsymbol{i}})^T \boldsymbol{\Psi}^{-1} (\widehat{\boldsymbol{x}}_{\boldsymbol{u}} - \widehat{\boldsymbol{y}}_{\boldsymbol{i}})\right), \tag{16}$$

where $\widehat{\boldsymbol{y}}_{\boldsymbol{i}} = (\hat{x}_i, \hat{y}_i)^T$ is the target position in the i-th frame, $\boldsymbol{\Psi}$ represents the shape of the target, and $\boldsymbol{\Phi} = (2\pi)^{\frac{1}{2}} \|\boldsymbol{\Psi}\|^{\frac{1}{2}}$. It is obvious that candidate pixel distribution is jointly determined by variables $\hat{\boldsymbol{x}}_{\boldsymbol{u}}, \widehat{\boldsymbol{y}}_{\boldsymbol{i}}$, and $\boldsymbol{\Psi}$. Therefore, the probability $p(\widehat{\boldsymbol{x}}_{\boldsymbol{u}})$ should be weighted by distribution model $N(\widehat{\boldsymbol{x}}_{\boldsymbol{u}}, \widehat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{\Psi})$,

$$\hat{p}(\widehat{\boldsymbol{x}}_{\boldsymbol{u}}) = N(\widehat{\boldsymbol{x}}_{\boldsymbol{u}}, \widehat{\boldsymbol{y}}_{\boldsymbol{i}}, \boldsymbol{\Psi}) * p(\widehat{\boldsymbol{x}}_{\boldsymbol{u}}), \tag{17}$$

and Equation (17) can be considered to be a sampling process of the distribution model using probability $\hat{p}(\widehat{\boldsymbol{x}}_{\boldsymbol{u}})$. Define

$$\psi = (\widehat{\boldsymbol{x}}_{\boldsymbol{u}} - \widehat{\boldsymbol{y}}_{\boldsymbol{i}})^T \boldsymbol{\Psi}^{-1} (\widehat{\boldsymbol{x}}_{\boldsymbol{u}} - \widehat{\boldsymbol{y}}_{\boldsymbol{i}}) \tag{18}$$

due to $\boldsymbol{\Psi}$ as a positive definite or semi-definite matrix, so that ψ can be described as follows:

$$\psi = (\widehat{\boldsymbol{x}}_{\boldsymbol{u}} - \widehat{\boldsymbol{y}}_i)^T \left(\boldsymbol{U}\boldsymbol{\Pi}\mathbf{U}^{\boldsymbol{T}}\right)^{-1} (\widehat{\boldsymbol{x}}_{\boldsymbol{u}} - \widehat{\boldsymbol{y}}_{\boldsymbol{i}}), \tag{19}$$

where $\boldsymbol{U} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}$ is a linear transformation and denotes the rotation transformation of the target, and $\Pi = \mathrm{diag}\left(\lambda_t^w, \lambda_t^h\right)$ represents the width and height of the target, respectively. By ignoring the rotation transformation of the target in the image plane in order to simplify computation, Equation (16) can be simplified as a generic ellipse formula,

$$\psi = (\hat{\boldsymbol{x}}_{\boldsymbol{u}} - \widehat{\boldsymbol{y}}_{\boldsymbol{i}})^T \boldsymbol{\Psi}^{-1} (\hat{\boldsymbol{x}}_{\boldsymbol{u}} - \widehat{\boldsymbol{y}}_{\boldsymbol{i}}) = \frac{(\hat{x} - \hat{x}_i)^2}{\lambda_t^w} + \frac{(\hat{y} - \hat{y}_i)^2}{\lambda_t^h} = \phi_t^s, \tag{20}$$

whose shape is determined by λ_t^w, λ_t^h, and ϕ_t^s, where ϕ_t^s denotes the Mahalanobis distance and is linearly dependent on the scale of the target.

When estimating the target size, it is necessary to traverse the search window to sample points

$$\lambda_t^w = \sqrt{\frac{\Sigma_{j=1}^n (\hat{x}_j - \hat{x}_i)^2}{n-1}}, \tag{21}$$

$$\lambda_t^h = \sqrt{\frac{\Sigma_{j=1}^n (\hat{y}_j - \hat{y}_i)^2}{n-1}}, \tag{22}$$

where $(\hat{x}_j, \hat{y}_j)$ denotes the sample point, and n is the number of sample points. The computational complexity is linearly related to the search window size. The tracking speed is significantly reduced in the case of large-scale targets. When the salient CN feature is applied, the back-projection is a sparse distribution. Therefore, a down-sampling strategy is reasonably adopted to sparsely sample the candidate pixels and use the sampling area to estimate the distribution in the entire search window. We perform down-sampling once to detect the target points sparsely in the search window. The sample interval can be selected according to the following empirical formula:

$$v = \max\left(\left[\frac{\min(\mathrm{w}, h)}{20}\right], 2\right), \tag{23}$$

and we can obtain the equal probability ellipse (only the major axis and the minor axis; the inclination angle is ignored). Using the definition of integral and the elliptic area formula we obtain

$$\phi_s^t \pi \sqrt{\lambda_t^w \lambda_t^h} \approx \int\int \delta(p(\hat{\boldsymbol{x}}_{\boldsymbol{u}}) - \xi) dx dy \approx v^2 \int\int \delta(p(\hat{\boldsymbol{x}}_{\boldsymbol{u}}) - \xi) d\frac{x}{v} d\frac{y}{v}, \tag{24}$$

where the ξ is the confidence threshold. Therefore,

$$\phi_s^t \approx v^2 \frac{\int\int \delta(p(\hat{\boldsymbol{x}}_{\boldsymbol{u}}) - \xi) d\frac{x}{v} d\frac{y}{v}}{\pi \sqrt{\lambda_t^w \lambda_t^h}}. \tag{25}$$

However, the application of salient features to back-projection inevitably leads to various probability distribution maps. That is, λ_t^w, λ_t^h do not represent the actual size of the target, so the estimated width and height should be scaled to the actual size $s_t = (w_t, h_t)$. We introduce the assumption of local and global consistency and the assumption that the target scale does not change drastically, so the size of the target at the t-th frame $s_t = (w_t, h_t)$ is obtained as follows:

$$\begin{cases} w_t = \frac{\lambda_t^w \sqrt{\phi_s^t}}{\lambda_{t-1}^w \sqrt{\phi_{t-1}^s}} * w_{t-1} \\ h_t = \frac{\lambda_t^h \sqrt{\phi_s^t}}{\lambda_{t-1}^h \sqrt{\phi_{t-1}^s}} * h_{t-1}. \end{cases} \tag{26}$$

To improve the stability of the scale estimation, we choose linear interpolation to update the target scale by

$$s_t^l = (1 - \mu) s_{t-1}^l + \mu s_t. \tag{27}$$

4.3. Tracking by Fusion

The estimated position and scale of the target can be computed by using the above two methods. However, the target position obtained by the standard kernel-based algorithm will be interfered by objects with similar color features in the background so as to result in a position shift. Changes in the target shape and the boundary effect will also cause a shift in the DCF method. Therefore, this paper adopts a particle fusion algorithm based on the Monte Carlo method to improve the accuracy of the estimated position and scale of the target.

The target appearance model in the search window can be expressed in the following form:

$$H^w(x) = \left[\begin{array}{cccc} h_1^w(x), & h_2^w(x), & \cdots, & h_n^w(x) \end{array} \right]. \tag{28}$$

We obtain two estimated position candidates via the proposed algorithm and KCF algorithm. The target position in the new frame can be expressed as a weighted sum of these two positions:

$$\bar{P} = \sum_{k=1}^{n} w_k P_k. \tag{29}$$

We use the Bhattacharyya distance between the appearance model in the search window and the target appearance model as the weights:

$$\rho = -\ln\left(\sum_{i=1}^{n} \sqrt{h_i^w(x) h_i^t(x)} \right). \tag{30}$$

By assuming that the interference caused by the background conforms to a Gaussian distribution, the weight can be thus computed as follows:

$$w_k = \frac{e^{\frac{(1-\rho_k)}{2\sigma^2}}}{\sum_{j=1}^{n} e^{\frac{(1-\rho_j)}{2\sigma^2}}}. \tag{31}$$

By updating the model at the target position in the new frame, we utilize the same online model updating strategy as other DCF-based trackers by:

$$h^t = (1-\eta)h^{t-1} + \eta h. \tag{32}$$

5. Experiments

To evaluate the performance of the proposed tracker, we implemented the proposed method with an Intel i5 2.50 GHz CPU, 16 GB RAM, and a Windows 10 × 64 operating system. The extensive experiments were conducted on the Temple-Color-128 (TC128) [29] dataset. TC128 is a dataset of 128-color video sequences that are used for visual tracking benchmarks. The dataset was created to study the role of color information in visual tracking. The dataset contains various challenging factors, such as occlusion, illumination change, deformation, motion blur, etc. The dataset also provides ground truth bounding boxes and attribute annotations for each sequence. We chose this dataset because it is a comprehensive and representative dataset for evaluating the performance of color-based trackers.

The performance of the algorithms was evaluated by: (i) average overlap ratio (AUC) of the success plot quantifying the result; (ii) the precision rate (PRE), which is the percentage of frames where the center distance between the predicted bounding box and the ground truth bounding box is within a certain threshold (we set it to 20 pixels); and (iii) the frames per second (FPS), which directly reflects the real-time performance of the algorithm. When evaluating the FPS of the algorithm, in order to simulate the real-time tracking scenario, we count the time for reading the image as the processing time.

5.1. Implementation Details

Our method was evaluated on TC128 by following the benchmark evaluation protocol [9]. All parameters of the algorithm were fixed for all video sequences.

For the PSACF, μ and η in Formulas (27) and (32) were set to 0.2 and 0.02, respectively. The confidence threshold ξ in Equation (24) was set to 0.25. Other parameters not specified were set the same as those for KCF.

5.2. Experiments and Results

5.2.1. Ablation Study

To verify the effectiveness of each component of the proposed algorithm, we conducted ablation studies with different modules enabled. Specifically, these modules include: (i) a scale estimation module (SE) based on a classic appearance model and probability estimation, where the color name feature is adopted as the color feature; (ii) down-sampling module (DS), where the sample interval is obtained by (23); and (iii) salient color feature filter module (SC), which uses the filtered color feature histogram instead of the classic background weighted histogram model.

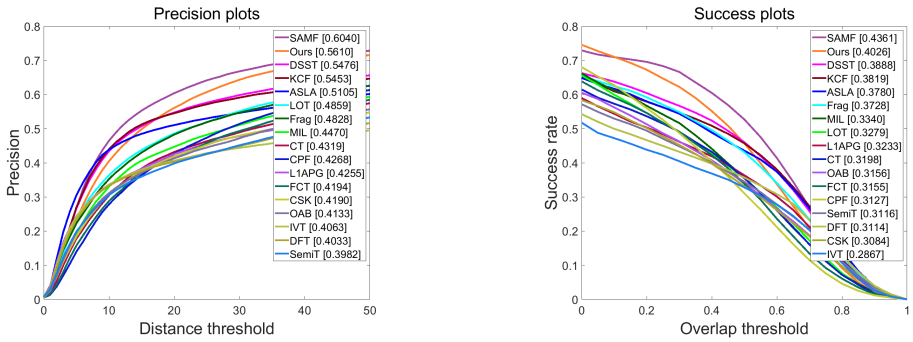
We conducted experiments on Temple-Color-128. The overall evaluation result is presented in Table 1. The baseline method without any components (NONE) has the lowest PRE and AUC. However, it also has the highest frame rate of 59.2853 FPS. The method with SE has a slightly higher PRE and AUC than the baseline method. The SE can improve the performance of tracking, but at the cost of computational efficiency. The results of the method with SE + DS show that DS can reduce the computational cost and increase the tracking speed, especially for large-scale targets, but at a cost of some accuracy loss. The method with all components enabled has the highest precision rate and average overlap ratio among all the methods. It also has a slightly lower frame rate than that of the method with scale estimation and down-sampling only. This shows that salient color name filters can effectively improve the performance, but at a small cost to computational efficiency. Overall, our method with all components SE + DS + SC has a superior performance in terms of accuracy and robustness compared to the other methods. This demonstrates that all components play an important role in the method.

Table 1. Ablation study of the proposed method. SE, DS, and SC represent scale estimation, down-sampling, and salient color name filters, respectively.

Components	PRE	AUC	FPS
NONE	0.5310	0.3388	59.2853
SE	0.5574	0.3415	40.3942
SE + DS	0.5408	0.3568	46.5514
SE + DS + SC	0.5610	0.4026	44.9217

5.2.2. Evaluation of Sequences

TC128 data are used for comparison among the proposed PSACF and other popular algorithms, as illustrated in Table 2. Figure 2 shows the success plot and precision plot. The proposed method achieves the second-best performance in terms of the AUC and PRE metric, which means that our method is more accurate and robust than the others. It also shows that our method can effectively handle various challenging factors, such as occlusion, illumination change, deformation, etc.



(**a**) Precision plot. (**b**) Success plot.

Figure 2. Evaluation of the TC128 dataset.

Table 2. Comparison experiment of the proposed method.

Methods	AUC	PRE	FPS
PSACF(Ours)	0.4026	0.5610	44.9217
SAMF	0.4361	0.6040	12.9108
DSST	0.3888	0.5476	15.5754
KCF	0.3819	0.5453	59.2853
CPT	0.3127	0.4268	4.1844
CT	0.3198	0.4319	8.9621
DFT	0.3114	0.4033	58.7570
FCT	0.3155	0.4194	35.2754
FragTrack	0.3728	0.4828	19.4681
IVT	0.2867	0.4063	8.3748
L1APG	0.3233	0.4255	2.9720
MIL	0.3340	0.4470	4.5498
OAB	0.3156	0.4133	37.5747

The proposed method also achieves a high frame rate of 44.9217 FPS, which is much faster than most of the other methods, except for KCF and DFT. This indicates that our method is more efficient and practical.

The proposed method has a slightly lower precision rate than SAMF. The DS module may introduce some errors in estimating the target position. However, this difference is not very significant, and the proposed method still has a high precision rate of 0.5610. This shows that our method can still provide reliable and consistent position estimates for the target.

Overall, the PSACF has a superior performance in terms of accuracy, robustness, efficiency, and speed compared to the other methods. This demonstrates the effectiveness and innovation of the proposed algorithm.

5.2.3. Performance Analysis

In Figure 3, some representative samples are given to analyze the performance of the proposed algorithm.

From the sequences Bag and Helicopter, it can be observed that our method can not only handle the scale variation but also adapt to the shape change of the target. When the target is non-rigidly deformed, the aspect ratio of the bounding box will vary accordingly. However, other algorithms are not sensitive to the nonrigid deformation of the target.

From the sequences Panda and Ball, it can be seen that the proposed algorithm can track fast-moving targets and is robust to motion blur and fast deformation. At the same time, the proposed algorithm will also reduce the influence of boundary effects.

From the sequences Tiger and Soccer1, the PSACF algorithm has superior long-term tracking abilities compared to other methods. Our scale estimation method can also locate the center of the salient features of the target, and the center is used to update the state. When the correlation filter tracker is affected by the boundary effect and deviates from the tracking target, we select the optimal results for integration according to the long-term target template maintained in memory.

Figure 4 shows some failure cases of the PSACF tracker. In the first row, the illumination changes drastically, which causes our appearance model to lose track of the target. The fusion tracking technique reduces the boundary effect if the illumination is stable. Additionally, if the scale of the target changes rapidly, especially when the illumination undergoes significant variation, our method will not be able to precisely estimate the size of the target. However, the SAMF, KCF, and DSST trackers can still track the target because they exploit the texture feature, which is insensitive to illumination change, for scale estimation. In the second and third rows, the target underwent complete occlusion for a few frames and several seconds, respectively. Every tracker could handle short-term occlusion but cannot deal with long-term complete occlusion, which also resulted in tracking failure.

In the fourth row, no tracker could handle tiny targets. In the last row, the target underwent non-rigid deformation in a short time span, which resulted in tracking failure.

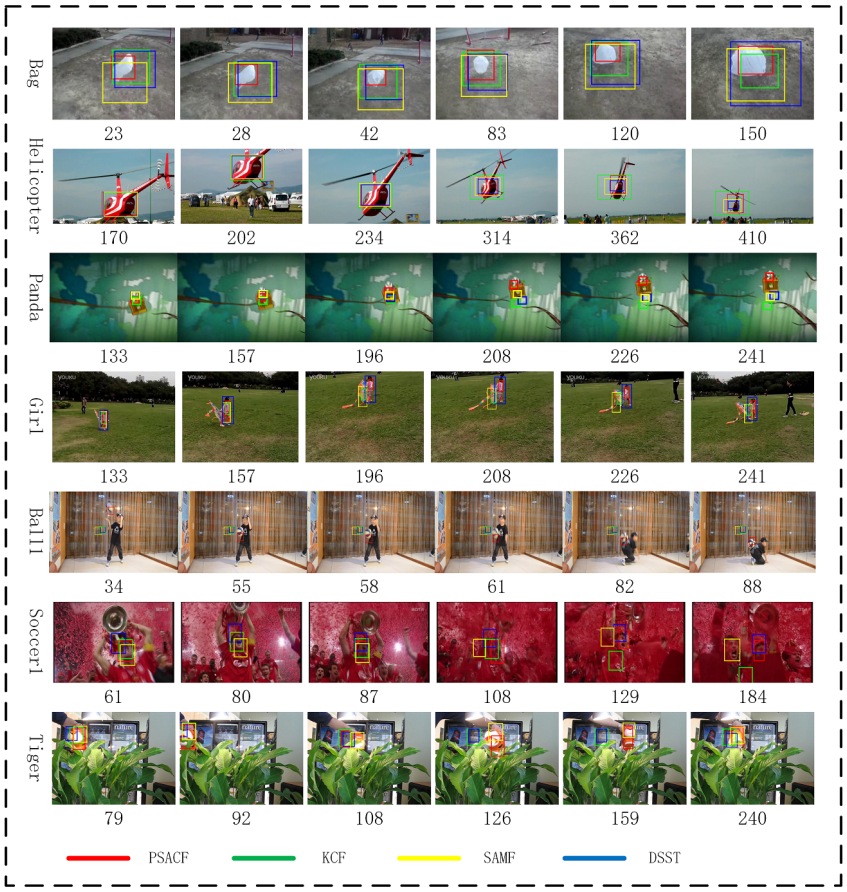


Figure 3. Illustration of the qualitative tracking results on challenging sequences.

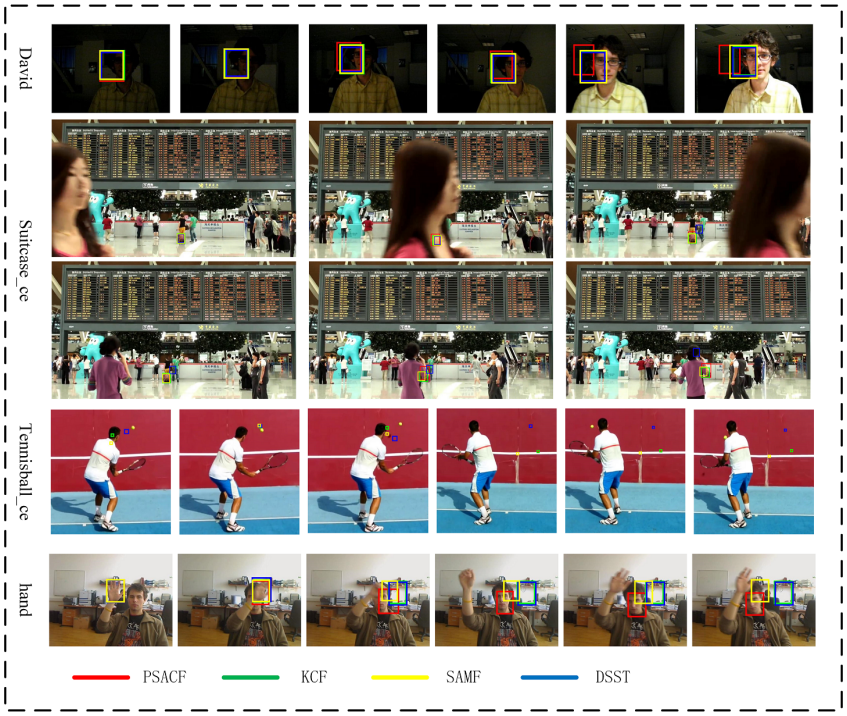


Figure 4. Some failure cases on challenging sequences.

6. Conclusions

In this paper, we addressed the problem of scale-adaptive tracking in computing power-constrained applications. We used color name (CN) features and a salient feature to reduce the target appearance model's dimensionality. We then estimated the target scale based on a Gaussian distribution model and introduced global and local scale consistency assumptions to restore the target's scale. We fused the tracking results with the DCF-based tracker to obtain the new position and scale of the target. Our research contributes a novel and efficient scale-adaptive tracking method that can be applied to various computing-constrained scenarios, such as embedded systems, edge computing, or mobile devices. However, our method still has some limitations and challenges that need to be addressed in future work. For example, our method may fail to track objects with long-term complete occlusion or in scenarios with drastic illumination changes. We plan to explore more robust latent information in target appearance methods, as well as more adaptive fusion strategies, to improve our method's performance in these challenging situations.

Author Contributions: Conceptualization, H.H.; methodology, H.H.; software, H.H.; validation, Z.C., Z.L., H.L. and X.L.; formal analysis, H.H., H.L. and Z.L.; investigation, H.H. and H.L.; resources, H.H. and Z.C.; data curation, H.H.; writing—original draft preparation, H.H.; writing—review and editing, Z.C., H.L., Z.L. and X.L.; visualization, H.H. and H.L.; supervision, Z.C. and X.L.; project administration, Z.C. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key R&D Program of China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: All authors informed consent.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Acknowledgments: The authors are grateful to Bangyu Li for his help with the preparation of resources in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bescos, B.; Campos, C.; Tardos, J.; Neira, J. DynaSLAM II: Tightly-Coupled Multi-Object Tracking and SLAM. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5191–5198. [CrossRef]
2. Xu, F.; Wang, H.; Chen, W.; Miao, Y. Visual Servoing of a Cable-Driven Soft Robot Manipulator With Shape Feature. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4281–4288. [CrossRef]
3. Fang, S.; Ma, Y.; Li, Z.; Zhang, B. A visual tracking algorithm via confidence-based multi-feature correlation filtering. *Multimed. Tools. Appl.* **2021**, *80*, 23963–23982. [CrossRef]
4. Rudenko, A.; Palmieri, L.; Doellinger, J.; Lilienthal, A.; Arras, K. Learning Occupancy Priors of Human Motion from Semantic Maps of Urban Environments. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3248–3255. [CrossRef]
5. Su, J.; He, X.; Qing, L.; Niu, T.; Cheng, Y.; Peng, Y. A novel social distancing analysis in urban public space: A new online spatio-temporal trajectory approach. *Sustain. Cities Soc.* **2021**, *68*, 102765. [CrossRef]
6. Ali, S.; Jonmohamadi, Y.; Takeda, Y.; Roberts, J.; Crawford, R.; Pandey, A. Supervised Scene Illumination Control in Stereo Arthroscopes for Robot Assisted Minimally Invasive Surgery. *IEEE Sens. J.* **2021**, *21*, 11577–11587. [CrossRef]
7. Cheng, T.; Li, W.; Ng, W.; Huang, Y.; Li, J.; Ng, C.; Chiu, P.; Li, Z. Deep Learning Assisted Robotic Magnetic Anchored and Guided Endoscope for Real-Time Instrument Tracking. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3979–3986. [CrossRef]
8. Fang, B.; Mei, G.; Yuan, X.; Wang, L.; Wang, Z.; Wang, J. Visual SLAM for robot navigation in healthcare facility. *Pattern Recognit.* **2021**, *113*, 107822. [CrossRef]
9. Wu, Y.; Lim, J.; Yang, M.H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef]
10. Danelljan, M.; Hager, G.; Khan, F.; Felsberg, M. Discriminative Scale Space Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1561–1575. [CrossRef] [PubMed]
11. Li, Y.; Zhu, J. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6–7 and 12, 2014, Proceedings, Part II 13;* Springer International Publishing: Cham, Switzerland, 2015; Volume 8926, pp. 254–265. [CrossRef]

12. Javed, S.; Danelljan, M.; Khan, F.S.; Khan, M.H.; Felsberg, M.; Matas, J. Visual Object Tracking With Discriminative Filters and Siamese Networks: A Survey and Outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 6552–6574. [CrossRef] [PubMed]
13. Henriques, J.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]
14. Han, R.Z.; Feng, W.; Wang, S. Fast Learning of Spatially Regularized and Content Aware Correlation Filter for Visual Tracking. *IEEE Trans. Image Process.* **2020**, *29*, 7128–7140. [CrossRef]
15. Piga, N.A.; Onyshchuk, Y.; Pasquale, G.; Pattacini, U.; Natale, L. ROFT: Real-Time Optical Flow-Aided 6D Object Pose and Velocity Tracking. *IEEE Robot. Autom. Lett.* **2022**, *7*, 159–166. [CrossRef]
16. Danelljan, M.; Khan, F.; Felsberg, M.; van de Weijer, J. Adaptive Color Attributes for Real-Time Visual Tracking. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1090–1097. [CrossRef]
17. Zuo, W.; Wu, X.; Lin, L.; Zhang, L.; Yang, M. Learning Support Correlation Filters for Visual Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1158–1171. [CrossRef] [PubMed]
18. Zhang, T.; Xu, C.; Yang, M. Learning Multi-Task Correlation Particle Filters for Visual Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 365–378. [CrossRef] [PubMed]
19. Zhang, L.; Suganthan, P.N. Robust visual tracking via co-trained Kernelized correlation filters. *Pattern Recognit.* **2017**, *69*, 82–93. [CrossRef]
20. Bolme, D.; Beveridge, J.; Draper, B.; Lui, Y. Visual Object Tracking using Adaptive Correlation Filters. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550. [CrossRef]
21. Ma, H.; Lin, Z.; Acton, S. FAST: Fast and Accurate Scale Estimation for Tracking. *IEEE Signal Process. Lett.* **2020**, *27*, 161–165. [CrossRef]
22. Lu, X.; Ma, C.; Ni, B.; Yang, X. Adaptive Region Proposal With Channel Regularization for Robust Object Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 1268–1282. [CrossRef]
23. Vojir, T.; Noskova, J.; Matas, J. Robust scale-adaptive mean-shift for tracking. *Pattern Recogn. Lett.* **2014**, *49*, 250–258. [CrossRef]
24. Chan, S.X.; Tao, J.; Zhou, X.L.; Bai, C.; Zhang, X.Q. Siamese Implicit Region Proposal Network With Compound Attention for Visual Tracking. *IEEE Trans. Image Process.* **2022**, *31*, 1882–1894. [CrossRef]
25. Li, S.J.; Zhao, S.; Cheng, B.; Chen, J.L. Dynamic Particle Filter Framework for Robust Object Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 3735–3748. [CrossRef]
26. Li, S.J.; Zhao, S.; Cheng, B.; Chen, J.L. Part-Aware Framework for Robust Object Tracking. *IEEE Trans. Image Process.* **2023**, *32*, 750–763. [CrossRef] [PubMed]
27. Yang, X.; Song, Y.; Zhao, Y.F.; Zhang, Z.S.; Zhao, C.Y. Unveil the potential of siamese framework for visual tracking. *Neurocomputing* **2022**, *513*, 204–214. [CrossRef]
28. van de Weijer, J.; Schmid, C.; Verbeek, J.; Larlus, D. Learning Color Names for Real-World Applications. *IEEE Trans. Image Process.* **2009**, *18*, 1512–1523. [CrossRef]
29. Liang, P.; Blasch, E.; Ling, H. Encoding Color Information for Visual Tracking: Algorithms and Benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5630–5644. [CrossRef] [PubMed]